$1.50
JULY, 1985
VOL.5 NO.6

M.A.C.E. JOURNAL

"Devoted Exclusively To The Atari Computer User"

IN THIS ISSUE...

ATARI LOGO
NEWS FROM CES
SOFTWARE REVIEWS
MACE BIRTHDAY PARTY
80-COLUMN BOARD REVIEW
DATAMATE - data storage in strings

Published by the Michigan Atari Computer Enthusiasts

FIRESIDE CHAT

Ok folks, get ready, here come the corrections to last month's column!

Dave Duberman from Atari finally agreed to come to the birthday party two days after the Journal went to the printer! It was nice to see that ST's really do exist! We all had a great time, Dave had a great time, and I would like to thank Dave and the powers that be at Atari for his attending!

Dave called again two days after the meeting and told me that we could take members' orders for the ST now! The only catch was it had to be done before this meeting. Gee, talk about a rush job. If you did get one, good luck! If not, fear not! ST's should be in the stores in about a month for the same price so there is not a big savings advantage. If you decided to wait, you may be better off! Just think, the pioneers out there will have to deal directly with Atari if something goes wrong. You'll have a local store to go back to! I hope with Atari's decision to sell ST's to user group members a real improvement in consumer relations is included.

What this all comes down to is this: MACE members were able to buy ST's because they belong to the club. The club placed a bulk order on behalf of its members. MACE is not responsible for the product you receive. You will have to deal directly with Atari should problems arise.

Movin' on, I didn't ramble in last month's hall of fame because I am not yet a Past President. But, I must say, I really have enjoyed it! 'Tis a great group we have here and I've made lots of new friends. My biggest goal was to bring a feeling of togetherness back to the group and, my gosh, I think it's working! I've only got two meetings left after this one (maybe, depending on September's election), but I have thoroughly enjoyed it!

In answer to a few questions concerning future Indus group purchases, I don't plan on it at this time. The original purchases offered a noticable savings to members. That is not the case anymore. The price difference between what the club could get them for now as opposed to Rite-Way, Just Software, etc. just isn't worth it. We're only talking about a $10.00 savings now. And besides, remember all those great door prizes we had at the birthday party.....

Till next month!

Kirk

^^*^*^*^*^*^*^*^*^*^*^*

COMING ATTRACTIONS

July 16th

MACE Library Disk Demos
SynCalc Demo - Paul Glass
MACE Awards Night

August 20th

MACE Swap Night
(contact Scott Garland for table reservations - members only!)

September 17th

Election of Officers

^^*^*^*^*^*^*^*^*^*^*^*

The MACE Journal is published monthly by the Michigan Atari Computer Enthusiasts. Unless otherwise noted, material published in the Journal is in the public domain and may be reproduced for private or user group use providing proper credit is given.

Submissions to the Journal can be mailed to the PO Box, uploaded to the MACE BBSs, any officer's BBS, or uploaded directly to the editor at 646-4455. Where possible, submissions should include a disk or tape file in AtariWriter or similar format and a working copy of the program. Specify format for screen dumps (AtariArtist, Koalapad, etc.). Authors whose submissions are published will receive a certificate good for a free disk or tape from the MACE library. Deadline for submissions is the first of each month.

DATAMATE:
Data Storage in Strings

by Ann McBain Ezzell

The Atari's character set has, with one exception, a printable character for every integer from 0 to 255. (The exception is 155, which corresponds to a carriage return.) This accounts for those funny-looking listings you sometimes see, with lines like:

```
100 A$="#$%&"
```

This may look like something out of a comic strip, but it is actually a compact representation of the numbers from 35 to 38. DATAMATE will create such a string for you from your data, then write string assignment statements like line 100 above and save them to disk or cassette for inclusion in another program. By taking advantage of the Atari's method of keeping track of variables, you can even put your data string into a specified location in memory.

Using strings to store data has the benefit of taking up less memory than the DATA statement method, as well as requiring less time than repetitive PEEKing and POKEing. Some of the more obvious applications include handling redefined character sets, machine language subroutines, player/missile graphics data, and custom display lists. You may be able to find other uses for the strings created by DATAMATE, but remember that only integer values from 0 to 255 can be stored in this way.

APPLICATIONS

The simplest use of DATAMATE is for loading machine language routines into a BASIC program. If you have a relocatable routine, make a string from it with DATAMATE and call it using:

```
100 X=USR(ADR(ML$))
```

You can of course also pass any needed parameters in the USR call. Non-relocatable routines can be stored with DATAMATE and POKEd into their proper locations:

```
100 FOR I=1 TO LEN(ML$)
110 POKE 1535+I,ASC(ML$(I))
120 NEXT I
130 X=USR(1536)
```

This would put your routine into page six.

Some of the other uses of DATAMATE require some knowledge of how the Atari stores and handles strings and other variables. As the Atari encounters each new variable, either as program lines are typed on the keyboard or ENTERed in from disk or cassette, the name is stored in the variable name table and eight bytes are allocated in the variable value table. Locations 130 and 131 hold the address of the beginning of the variable name table (PEEK(130) + 256*PEEK(131) will give you the VNT address); 134 and 135 hold the address of the variable value table (VVT).

Your Atari uses three types of variables: scalar (simple numeric values), array, and string. Each type of variable uses its eight bytes in the VVT in a different way; here we are only concerned with string variables. The first byte indicates the type of variable. 129 and 128 respectively indicate dimensioned and undimensioned strings. Byte 2 shows the variable number, from 0 to 127. The third and fourth bytes combine (low byte/high byte) to tell you the location of the string in terms of its offset from the string and array table, which starts at the end of the BASIC program. (Locations 140 and 141, labelled STARP, hold the address of this table.) Bytes 5 and 6 hold the actual length of the string and the last two bytes hold its dimensioned length.

The structure of the variable value table gives you the power to manipulate strings for special uses. You can force a string to a chosen location and specify its actual and dimensioned lengths. You have to know how far into the table the eight bytes for your string are, so the easiest thing is to make your string the first variable in the table. You can do this by having that string be the first variable of any type referenced in your program. If you have been working on a program for a while, put in a line dimensioning your string before any other variables are mentioned, LIST the program to disk or cassette, type NEW, and reENTER the

program. Variables will be entered into the VNT and VVT in the order in which they appear within the program. (When you SAVE a program, the VNT and VVT are saved along with the tokenized program. LISTing saves only the program itself, in untokenized form.)

Listing 2 shows one way of setting up a string in memory to hold a redefined character set. The full character set for the Atari takes up 1024 bytes of memory - 128 characters times 8 bytes per character. I will assume that you have generated the data to define a new character set and have used DATAMATE to create a string called DAT$ to hold the data. Now you need a safe place to store your data and some way to put it there.

Line 100 dimensions DAT$, making it the first entry in the variable table. Since you will be moving DAT$ and setting aside 1K of memory for it elsewhere, you can dimension it to one byte initially and not waste space in the string and array table.

One reasonably safe location for your new character set is above what your Atari thinks is the top of memory. Location 106, RAMTOP, holds the number of pages (1 page = 256 bytes) of available memory. You can change the value in RAMTOP and your Atari will more or less leave the area above it alone. You do have to be careful, though, because certain actions will disturb some of the memory above RAMTOP. Issuing a GRAPHICS or "CLEAR SCREEN" (PRINT CHR$(125)) command will clear out the first 64 bytes above RAMTOP. Scrolling the text window wipes out up to 800 bytes. It is best to leave a buffer zone unless you are certain that your program will not do anything to interfere with the area above RAMTOP.

Full character sets must start on a 1K boundary (i.e., an address evenly divisible by 1024, or a page number divisible by 4). Half sets, like those used in GR.1 and GR.2, can start on a 1/2 K boundary. Since this example deals with a full character set, line 110 moves RAMTOP down by eight pages (four pages for the character set plus a four page safety zone). This puts the start of the character set on a 1K boundary.

Lines 120 and 130 define VT and AT as the addresses of the variable value table and the string and array table. Line 140 sets OS equal to the offset between the start of the character set (remember that it is four pages above the new RAMTOP) and the string and array table. The value of OS is broken down into a high byte and a low byte in line 150.

Next, the new values are POKEd into the variable table. You do not want to change the first two bytes in the table. The third and fourth bytes receive the offset for the string's new location, the next two the new actual length (0 + 256*4 = 1024), and the last two get the new dimensioned length. If you were writing your own program, you would now be ready to put in the string assignment statements. (Remember? That's what DATAMATE is for.)

In this example, just to prove to you that this really does work, I have filled DAT$ with A's, which should put 65's into 1024 bytes starting four pages above the new RAMTOP. (65 is the ATASCII value of "A".) Line 200 looks into this area of memory and prints out the contents for you.

After changing RAMTOP, you must execute a graphics command so that the display list and screen memory area will be moved below the new RAMTOP. If you are using this technique to set up a new character set, you will have to POKE 756 with the page number of your character set, so that ANTIC will fetch the character data from there instead of the normal ROM character set.

Now you know how to put a string where you want it to be. You will find that this method is much faster than looping through DATA statements and POKEing each number into its target location. You can also use this string manipulation technique to set up an area for player/missile graphics, with a large blank string encompassing the PMG area and small strings which hold your player data moving around as substrings within the larger one. Non-relocatable machine language routines can be written to reside above a lowered RAMTOP or some other "safe" area and handled in the same way. You could also store a custom display list in a string and force it to its

proper location. Unfortunately, you cannot place strings in page 6 with this method, because the offset from the string and array table would have a negative value.

If you need to specify the location of more than one string, dimension the strings you need at the beginning of your program and treat each one as in Listing 2. Remember that you will use VT+10 through VT+15 for the second string, VT+18 through VT+23 for the third, and so on.

USING DATAMATE

Type in Listing 1 and SAVE it. If you want to use DATAMATE to make strings from DATA statements in existing programs, you should also store it using the LIST command so that you can merge it with your program. To make strings from a binary file or from keyboard entries, RUN DATAMATE and follow the screen directions. To make strings from DATA statements, load your "host" program, make sure that there are no lines that will overlap with DATAMATE, ENTER DATAMATE, and GOTO 30000. You will need to know at what line to start reading the data and how many items to read.

Your first choice when using DATAMATE will be the method of data entry: DATA statements, binary file, or keyboard entry. If you choose the keyboard, you may use decimal or hexadecimal format. Next you must specify the name of the string which will hold your data. Remember that only upper-case letters and numbers are allowed in string names, and that the first character must be a letter. DATAMATE limits your string names to 10 characters in length. You can include the "$" character at the end or not, as you wish.

If you want to read the instructions, press "Y"; otherwise press "N" to continue. Depending on the method you choose, you will have to answer certain questions about your data. For DATA statements, give the line number of the first DATA statement and the number of items. The binary file method will ask you for a filename; be sure to specify the device (cassette or disk drive 1). For entry from the keyboard, you must give the number of items. With all methods, DATAMATE will then ask for the line number to begin string storage and the line number increment to use. Be sure to choose values that will not overwrite line 30000, the beginning of DATAMATE.

You will have a chance to correct your entries, then the program will begin getting the data. The first two methods proceed on their own; for the third you must enter items from the keyboard one at a time. Use two characters for every hexadecimal entry. When you have entered the specified number of items, you will have a chance to correct any mistakes you may have noticed. You can choose to reenter all of the data or only selected items.

Once all the data has been converted to string form, the program will write string assignment statements. The screen will go blank during this part of the program to speed it up. If the screen is blank for more than a short time, the internal speaker will click occasionally to assure you that your computer has not gone to sleep. The display will return and you can see the newly written lines. DATAMATE can LIST the lines to cassette or disk so that they can be ENTERed into your program as needed.

SPECIAL PROGRAM TECHNIQUES

As I mentioned above, the Atari has no printing character which corresponds to the number 155. DATAMATE will note and replace any values of 155 in your data in such a way that the actual value will be put back before the data is used. Another data value which can cause trouble is 34, which prints out as quotes ("). Since quotation marks are used as string delimiters, trying to include them as part of a string will cause an error. Values of 34 are also replaced with dummy values. DATAMATE takes care of this is in a subroutine starting at line 31000.

Each time that a value of 34 or 155 is found in the data, FLAG34 or FLAG155 is incremented. This not only notes that one of those values has been found, it determines the dimension of the array used to hold its location in the string. (Two elements are needed for each entry in the array to allow for string indices

greater than 255.) The routine starting at line 31000 dimensions the arrays HOLD34 and HOLD155 as needed, then replaces the 34's and 155's with dummy values and fills the arrays with the locations of the replaced characters. Lines 30400-30595 write program lines to put the correct values back into the string like this:

```
100 DAT$(2,2) = CHR$(34): DAT$(45,45) = CHR$(34)
```

The new program lines are all written using Atari's "forced read" mode, which acts as if the RETURN key were being pushed continually. See lines 30310-30330 for an example. Location 842 holds auxiliary byte number one (AUX1) for I/O Control Block (IOCB) zero. IOCB zero is normally used for the screen editor (E:). A value of 13 POKEd into 842 sets the screen up for input as well as output.

If you want to use the "forced read" mode in your own programs, clear the screen and position the cursor at least four lines down from the top. Print whatever program lines you want to be entered, then print "CONT" (for continue). Position the cursor at the top of the screen, POKE 842,13 and STOP the program. The "forced read" mode will go into effect and read down the screen, entering each line as if typed in from the keyboard, until it reaches the CONT command. It will then continue the program starting with the first statement on the physical program line following the line where the STOP command was. POKE 842,12 to get back to the regular screen editor mode (keyboard input/screen output) and repeat as many times as necessary. The process is interesting, if rather dizzying, to watch, but will happen about 30% faster if you turn off ANTIC (POKE 559,0) while it is happening. POKE 559,34 to turn the screen back on.

PROGRAM OUTLINE

30000-30285
Initial screens to determine method of DATA entry and provide instructions

30290-30640
Write string assignment statements and enter them using Atari "forced read" mode

30700-30780
Save new lines to disk drive 1 or cassette

31000-31040
Replace values of 34 and 155 with dummy values and place location of dummy values into arrays for later reinsertion of actual values

31100-31180
Read values from DATA statements and convert to string form

31200-31290
Read values from binary file and convert to string form

31300-31495
Input values from keyboard and convert to string form

31360-31390
Correct values in string from keyboard entry

31400-31410
Subroutine for decimal entries from keyboard

31420-31435
Subroutine for hexadecimal entries from keyboard

31500-31550
Error handling

31600
Get single value from keyboard

31700-31740
Get information for string assignment statements

31800-31850
Check validity of filename

I think that you will be able to find many uses for DATAMATE, both with programs you write yourself and those you find in magazines (they are notorious for using DATA statements instead of strings). Good luck, and enjoy the program.

```
30000 REM DATAMATE Listing 1
30010 CLR :DIM BF$(14),D$(3),NAME$(11)
:POKE 710,146
30015 ? "{CLEAR}{DOWN}{TAB}{TAB}DATAMA
TE":? "{DOWN}Create a string from:":?
"{DOWN}{TAB}1) DATA statements":? "{DO
WN}{TAB}2) Binary file"
30020 ? "{DOWN}{TAB}3) Keyboard entry"
:? "{DOWN}Select desired method of inp
ut:";:OPEN #1,4,0,"K:"
30025 GET #1,M:IF M<49 OR M>51 THEN 30
025
30030 M=M-48:? CHR$(M+176);
30035 METHOD=31000+100*M:IF M<>3 THEN
CLOSE #1:GOTO 30055
30040 ? :? "{DOWN}{DOWN}Entries in (1)
 Decimal":? "{DOWN}{TAB}{TAB}{LEFT}{LE
FT}(2) Hex{TAB}";
30045 GET #1,NUM:IF NUM<49 OR NUM>50 T
HEN 30045
30050 CLOSE #1:NUM=NUM-49:? CHR$(NUM+1
77)
30055 ? :? :? "Name of string (up to 1
0 characters)"
30060 INPUT NAME$:IF LEN(NAME$)=0 THEN
 ? "{UP}";:GOTO 30060
30065 IF ASC(NAME$)<65 OR ASC(NAME$)>9
0 THEN ? :? "{BELL}FIRST CHARACTER MUS
T BE UPPER-CASE A-Z":? "{UP}{UP}";:GOT
O 30055
30070 L=LEN(NAME$):IF ASC(NAME$(L))=36
 THEN L=L-1:STRFLAG=1:REM $ last chara
cter
30075 FOR I=1 TO L:A=ASC(NAME$(I))
30080 IF A<48 OR (A>57 AND A<65) OR A>
90 THEN ? :? "{BELL}ONLY A-Z AND 0-9 A
LLOWED IN NAMES";:GOTO 30055
30085 NEXT I:IF  NOT STRFLAG THEN NAME
$(L+1)="$"
30087 POKE 752,1:? :? :? "  DO YOU WAN
T INSTRUCTIONS? (Y/N)";
30090 GOSUB 31600:IF A<>89 AND A<>78 T
HEN 30090
30095 IF A=78 THEN 30285
30100 ? "{CLEAR}{DOWN}DATAMATE will cr
eate a string for data";:? "storage fr
om DATA statements, binary"
30110 ? "files, or keyboard entries in
 decimal":? "or hexadecimal format.  O
NLY integer"
30120 ? "values from 0 to 255 are allo
wed.":? :? "When choosing a name for y
our string,"
30130 ? "avoid using BASIC keywords.
You may":? "use a name of up to 10 cha
racters.":?
30140 ? "The starting line number for
string":? "storage must be chosen so t
hat the"
30150 ? "new lines will not overwrite
the":? "beginning of this program at l
ine"
30160 ? "30000.  Also be sure not to c
hoose":? "line numbers that will inclu
de lines"
30170 ? "already present if you are re
ading":? "from DATA statements in a pr
ogram."
30180 ? :? "Use only D: or C: as devic
e names.":? "{DOWN}{TAB}PRESS ANY KEY
TO CONTINUE";:POKE 764,255:GOSUB 31600
30190 ? "{CLEAR}{DOWN}If you have any
values of 34 or 155":? "in your data,
this program will"
30200 ? "replace them with dummy value
s and":? "write assignment statements
to put"
30210 ? "the correct values back into
the":? "string when it is used."
30220 ? :? "This is necessary because
CHR$(34)":? "is quotes (";CHR$(34);"),
 which will cause an"
30230 ? "error when encountered in the
 middle":? "of a string.":? :? "CHR$(1
55) is a carriage return,"
30240 ? "which will not print properly
.":? :? "If you want to print out your
 string,"
30250 ? "you may have to POKE 766,1 so
 that":? "any non-printing characters
like":POKE 766,1
30260 ? "{UP} and {DOWN} will print pr
operly.":? "POKE 766,0 to return to no
rmal mode.":POKE 766,0
30280 ? "{DOWN}{TAB}PRESS ANY KEY TO C
ONTINUE";:POKE 764,255:GOSUB 31600
30285 POKE 752,0:GOSUB METHOD:SUBSTRS=
INT(STRLEN/80)
30290 REM write string assignment stat
ements
30295 ? "{CLEAR}{DOWN}The screen will
go blank briefly":? "while the string
assignment statements":? "{UP}are bein
g written."
30300 ? "{DOWN}{DOWN}{TAB}PRESS ANY KE
Y TO CONTINUE":GOSUB 31600:TRAP 31500:
POKE 559,0:IF FLAG34 OR FLAG155 THEN G
OSUB 31000
```

```
30305 ELN=SLN:IF SUBSTRS=0 THEN I=0:GO
TO 30335
30310 FOR I=0 TO SUBSTRS-1:? "{CLEAR}{
DOWN}{DOWN}{DOWN}{DOWN}":POKE 766,1:PO
KE 53279,0
30320 ? ELN;" ";NAME$;"(";I*80+1;",";I
*80+80;")=";CHR$(34);DAT$(I*80+1,I*80+
80):POKE 766,0:? "CONT"
30325 POSITION 2,0:POKE 842,13:STOP
30330 POKE 842,12:ELN=ELN+LNINC:NEXT I
30332 IF SUBSTRS*80=LEN(DAT$) THEN 303
60
30335 ? "{CLEAR}{DOWN}{DOWN}{DOWN}{DOW
N}"
30340 POKE 766,1:? ELN;" ";NAME$;"(";I
*80+1;")=";CHR$(34);DAT$(I*80+1):POKE
766,0:? "CONT"
30345 POSITION 2,0:POKE 842,13:STOP
30350 POKE 842,12:ELN=ELN+LNINC
30360 IF FLAG34=0 AND FLAG155=0 THEN 3
0600
30400 REM write lines to correct chang
es for values of 34 and 155
30410 IF FLAG34=0 THEN 30510
30415 LINES=INT(FLAG34/3):IF LINES=0 T
HEN J34=1:GOTO 30480
30420 FOR J34=0 TO LINES-1:? "{CLEAR}{
DOWN}{DOWN}{DOWN}{DOWN}":POKE 53279,0
30425 ? ELN;:FOR K34=J34*6+1 TO J34*6+
6 STEP 2:IT=HOLD34(K34)+256*HOLD34(K34
+1)
30427 ? NAME$;"(";IT;",";IT;")=CHR$(34
):";:NEXT K34:? "{BACK S}":? "CONT"
30430 POSITION 2,0:POKE 842,13:STOP
30435 POKE 842,12:ELN=ELN+LNINC:NEXT J
34:J34=6*LINES+1
30440 IF LINES*3=FLAG34 THEN 30500
30480 ? "{CLEAR}{DOWN}{DOWN}{DOWN}{DOW
N}":POKE 53279,0
30485 ? ELN;:FOR K34=J34 TO FLAG34*2 S
TEP 2:IT=HOLD34(K34)+256*HOLD34(K34+1)
30487 ? NAME$;"(";IT;",";IT;")=CHR$(34
):";:NEXT K34:? "{BACK S}":? "CONT"
30490 POSITION 2,0:POKE 842,13:STOP
30495 POKE 842,12:ELN=ELN+LNINC
30500 IF FLAG155=0 THEN 30600
30510 LINES=INT(FLAG155/3):IF LINES=0
THEN J55=1:GOTO 30580
30520 FOR J55=0 TO LINES-1:? "{CLEAR}{
DOWN}{DOWN}{DOWN}{DOWN}":POKE 53279,0
30525 ? ELN;:FOR K55=J55*6+1 TO J55*6+
6 STEP 2:IT=HOLD155(K55)+256*HOLD155(K
55+1)
30527 ? NAME$;"(";IT;",";IT;")=CHR$(15
5):";:NEXT K55:? "{BACK S}":? "CONT"
30530 POSITION 2,0:POKE 842,13:STOP
30535 POKE 842,12:ELN=ELN+LNINC:NEXT J
55:J55=6*LINES+1
30540 IF LINES*3=FLAG155 THEN 30595
30580 ? "{CLEAR}{DOWN}{DOWN}{DOWN}{DOW
N}":POKE 53279,0
30585 ? ELN;:FOR K55=J55 TO FLAG155*2
STEP 2:IT=HOLD155(K55)+256*HOLD155(K55
+1)
30587 ? NAME$;"(";IT;",";IT;")=CHR$(15
5):";:NEXT K55:? "{BACK S}":? "CONT"
30590 POSITION 2,0:POKE 842,13:STOP
30595 POKE 842,12:ELN=ELN+LNINC
30600 ELN=ELN-LNINC
30610 POKE 559,34:? "{CLEAR}{DOWN}";NA
ME$;" complete and listed from":? "lin
e ";SLN;" to line ";ELN;"."
30620 ? "{DOWN}{DOWN}   PRESS ANY KEY
TO SEE NEW LINES":GOSUB 31600
30640 POKE 766,1:LIST SLN,ELN:POKE 766
,0
30700 ? :? "{TAB}   SAVE NEW LINES (Y/
N)?"
30710 GOSUB 31600:IF A<>89 AND A<>78 T
HEN 30710
30720 IF A=78 THEN POKE 752,0:END
30730 ? "{CLEAR}{DOWN}{TAB}   SAVE CRE
ATED STRING":?
30740 POKE 752,0:? :? "LIST to what fi
le";:INPUT BF$:GOSUB 31800
30750 POKE 752,1:? :? "Prepare storage
 device.":? :? "{TAB}PRESS ANY KEY TO
CONTINUE";:GOSUB 31600
30760 BACK=30740:TRAP 31500:LIST BF$,S
LN,ELN:POKE 752,0
30770 IF BF$(1,2)="C:" THEN ? :? "Stri
ng LISTed to cassette.":END
30780 ? :? "String LISTed to disk as:"
:? "{TAB}";BF$:END
31000 REM replace 34's and 155's
31010 DIM HOLD34(2*FLAG34),HOLD155(2*F
LAG155):J34=1:J155=1
31020 FOR I=1 TO LEN(DAT$):IHI=INT(I/2
56):ILO=I-IHI*256
31025 IF ASC(DAT$(I))=34 THEN HOLD34(J
34)=ILO:HOLD34(J34+1)=IHI:J34=J34+2:DA
T$(I,I)="x"
31030 IF ASC(DAT$(I))=155 THEN HOLD155
(J155)=ILO:HOLD155(J155+1)=IHI:J155=J1
55+2:DAT$(I,I)="y"
31040 NEXT I:RETURN
31100 REM create string from DATA stat
ements
```

```
31105 ? "{CLEAR}{DOWN}CREATE STRING FR
OM DATA STATEMENTS":?
31110 TRAP 31110:? :? "First line of D
ATA statements";:INPUT DLN
31115 TRAP 31115:? :? "Number of items
";:INPUT STRLEN:GOSUB 31700
31120 TRAP 31175:RESTORE DLN
31125 FOR I=1 TO STRLEN:READ D:IF D<0
OR D>255 THEN 31180
31135 DAT$(I)=CHR$(D):IF D=34 THEN FLA
G34=FLAG34+1
31165 IF D=155 THEN FLAG155=FLAG155+1
31170 NEXT I:RETURN
31175 ? :? "{BELL}OUT OF DATA AT ";I-1
;" ITEMS{DOWN)":? "PLEASE CHECK DATA S
TATEMENTS":POKE 752,0:END
31180 ? :? "{BELL}ITEM # ";I;" IS NOT
WITHIN 0-255{DOWN}":? "PLEASE CHECK DA
TA STATEMENTS":POKE 752,0:END
31200 REM create string from binary fi
le
31205 ? "{CLEAR}{DOWN}  CREATE STRING
FROM BINARY FILE":?
31210 ? :? "Binary filename":? "   (in
cluding device)";:INPUT BF$:GOSUB 3180
0
31220 BACK=31210:TRAP 31500:OPEN #2,4,
0,BF$:GET #2,A:GET #2,B
31225 IF A<>255 OR B<>255 THEN POKE 75
2,1:? :? "{BELL}NOT A BINARY FILE":CLO
SE #2:GOTO 31210
31230 GET #2,LOST:GET #2,HIST:GET #2,L
OEN:GET #2,HIEN
31235 STRLEN=HIEN*256+LOEN-HIST*256-LO
ST+1
31240 ? :? "Length of string will be:
";STRLEN:GOSUB 31700:TRAP 31500
31280 FOR I=1 TO STRLEN:GET #2,D:DAT$(
I)=CHR$(D):IF D=34 THEN FLAG34=FLAG34+
1
31285 IF D=155 THEN FLAG155=FLAG155+1
31290 NEXT I:CLOSE #2:RETURN
31300 REM create string from keyboard
entries
31305 ? "{CLEAR}{DOWN}CREATE STRING FR
OM KEYBOARD ENTRIES":?
31310 TRAP 31310:? :? "Number of entri
es";:INPUT STRLEN:GOSUB 31700
31315 ? "Enter data one at a time:":?
:POKE 752,0:ENTRY=31400+20*NUM
31320 FOR I=1 TO STRLEN:GOSUB ENTRY
31325 DAT$(I)=CHR$(D):IF D=34 THEN FLA
G34=FLAG34+1
31330 IF D=155 THEN FLAG155=FLAG155+1
31335 NEXT I:POKE 752,1:? :? "{TAB}{TA
B}LAST ENTRY"
31340 ? "{TAB}ALL ENTRIES CORRECT? (Y/
N)";
31345 GOSUB 31600:IF A=89 THEN RETURN
31350 IF A=78 THEN 31360:REM correct e
ntries
31355 GOTO 31345
31360 POKE 752,0:? :? :? "Correct:":?
"{TAB}(1) All entries":? "{TAB}(2) Sel
ected entries  ";
31365 GOSUB 31600:IF A<49 OR A>50 THEN
 31365
31370 ? CHR$(A+128):? :IF A=49 THEN 31
315
31375 TRAP 31375:? :? "Which entry # t
o correct";:INPUT I:IF I<1 OR I>STRLEN
 THEN 31480
31380 ? :GOSUB ENTRY:DAT$(I,I)=CHR$(D)
:IF D=34 THEN FLAG34=FLAG34+1
31385 IF D=155 THEN FLAG155=FLAG155+1
31390 ? :GOTO 31340
31400 TRAP 31485:? "Entry # ";I;" ";:I
NPUT D:IF D<0 OR D>255 THEN 31485
31410 RETURN
31420 TRAP 31490:? "Entry # ";I;" ";:I
NPUT D$:IF LEN(D$)<>2 THEN 31490
31425 D1=ASC(D$):D2=ASC(D$(2))
31430 IF D1<48 OR (D1>57 AND D1<65) OR
 D1>70 OR D2<48 OR (D2>57 AND D2<65) O
R D2>70 THEN 31495
31435 D1=(D1-48)*(D1<58)+(D1-55)*(D1>6
4):D2=(D2-48)*(D2<58)+(D2-55)*(D2>64):
D=16*D1+D2:RETURN
31480 ? :? "{BELL}ENTRIES FROM 1 TO ";
STRLEN;" ONLY":GOTO 31375
31485 ? :? "{BELL}NUMBERS ONLY 0-255 *
CORRECT LAST ENTRY":GOTO 31400
31490 ? :? "{BELL}TWO CHARACTERS ONLY*
CORRECT LAST ENTRY":GOTO 31420
31495 ? :? "{BELL}HEX FORMAT 0-9,A-F *
CORRECT LAST ENTRY":GOTO 31420
31500 REM error checking
31510 POKE 559,34:E=PEEK(195)
31520 ? "{BELL}":POKE 752,1:CLOSE #2:I
F E=170 THEN ? "FILE NOT FOUND":TRAP 3
1500:GOTO BACK
31525 IF E=162 THEN ? "DISK FULL":TRAP
 31500:GOTO BACK
31530 IF E=130 THEN ? "UNKNOWN DEVICE"
:TRAP 31500:GOTO BACK
31535 IF E=165 THEN ? "BAD FILE NAME":
TRAP 31500:GOTO BACK
31540 IF E=138 THEN ? "DEVICE DOES NOT
```

```
 RESPOND":TRAP 31500:GOTO BACK
31545 IF E=167 THEN ? "FILE LOCKED - U
NABLE TO WRITE":TRAP 31500:GOTO BACK
31550 ? "ERROR NUMBER ";E;" AT LINE ";
PEEK(186)+256*PEEK(187):END
31600 OPEN #1,4,0,"K:":GET #1,A:CLOSE
#1:RETURN
31700 TRAP 31700:? :? "First line for
string storage";:INPUT SLN
31710 TRAP 31710:? :? "Line # incremen
ts";:INPUT LNINC:?
31720 TRAP 40000:POKE 752,1:? :? :? "{
TAB}ALL ENTRIES CORRECT? (Y/N)";
31730 GOSUB 31600:IF A=78 THEN POP :CL
OSE #2:POKE 752,0:GOTO 30285
31735 IF A<>89 THEN 31730
31740 ? "{CLEAR}{DOWN}Creating string.
..":DIM DAT$(STRLEN):FLAG34=0:FLAG155=
0:RETURN
31800 REM check filename
31810 IF BF$="C:" THEN RETURN
31820 IF LEN(BF$)<3 THEN 31850
31830 IF BF$(1,2)<>"D:" THEN 31850
31840 RETURN
31850 ? :? "{BELL}SPECIFY C: OR D:FILE
NAME":? :INPUT BF$:GOTO 31810
```

```
0 REM Listing 2 - Variable manipulatio
n demo
100 DIM DAT$(1)
110 POKE 106,PEEK(106)-8:RAMTOP=PEEK(1
06)
120 VT=PEEK(134)+256*PEEK(135)
130 AT=PEEK(140)+256*PEEK(141)
140 OS=(RAMTOP+4)*256-AT
150 HI=INT(OS/256):LO=OS-256*HI
160 POKE VT+2,LO:POKE VT+3,HI
170 POKE VT+4,0:POKE VT+5,4
180 POKE VT+6,0:POKE VT+7,4
190 DAT$="A":DAT$(1024)="A":DAT$(2)=DA
T$
200 GRAPHICS 0:FOR I=0 TO 1023:? PEEK(
256*(RAMTOP+4)+I);" ";:NEXT I
```

MACE JOURNAL LISTING CONVENTIONS

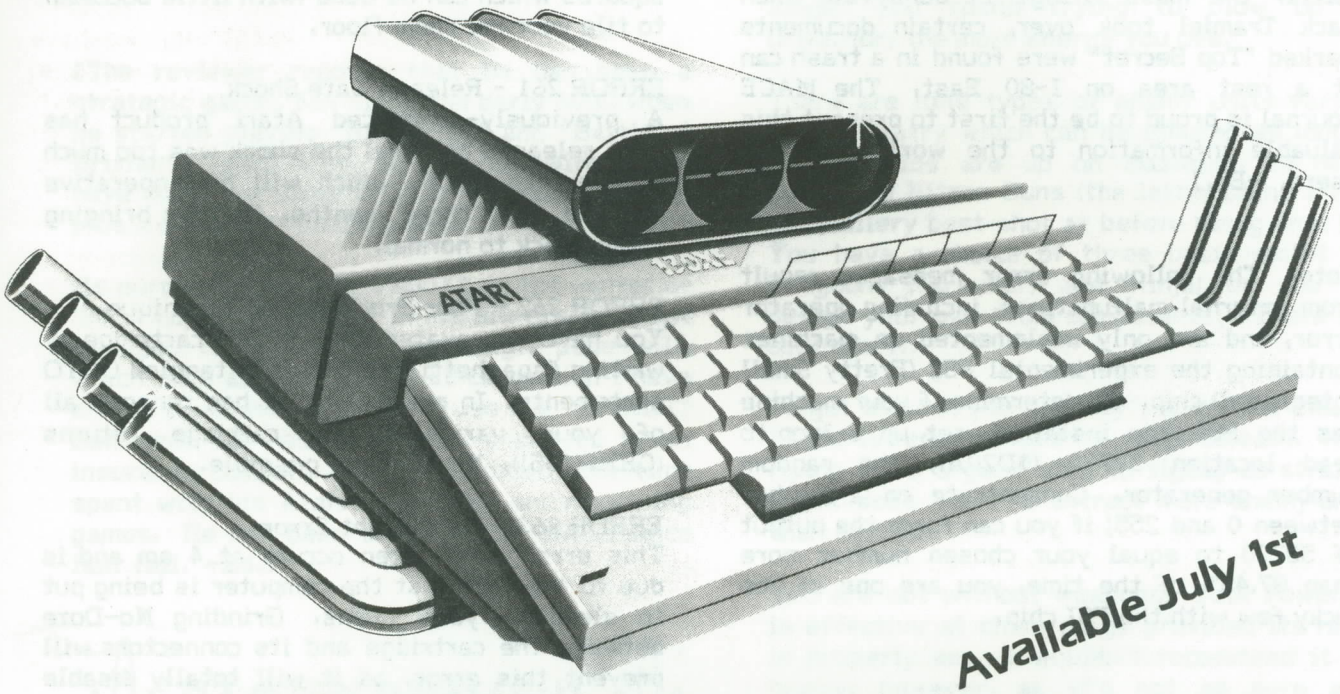
To reduce our readers' eyestrain, we have adopted a special method for listing programs. Programs will be listed in 38 column format, and certain characters will be replaced by an abbreviated form of their function, printed within curly braces (see below). Any characters to be typed in inverse video will be underlined, and control characters will be represented by their respective letters within curly braces. If a character within braces is also underlined, toggle the inverse video on and then hold down the control key while typing the character.

This method may seem awkward at first, but you should quickly get used to it, and the listings will be much easier to read. The special characters which will be spelled out are as follows:

When you see:	You should type:
{CLEAR}	ESC SHIFT <
{UP}	ESC CTRL -
{DOWN}	ESC CTRL =
{LEFT}	ESC CTRL +
{RIGHT}	ESC CTRL *
{BACK S}	ESC DELETE
{DELETE}	ESC CTRL DELETE
{INSERT}	ESC CTRL INSERT
{DEL LINE}	ESC SHIFT DELETE
{INS LINE}	ESC SHIFT INSERT
{TAB}	ESC TAB
{CLR TAB}	ESC CTRL TAB
{SET TAB}	ESC SHIFT TAB
{BELL}	ESC CTRL 2
{ESC}	ESC ESC
{COMMA}	CTRL , (comma)
{PERIOD}	CTRL . (period)
{SEMI-COLON}	CTRL ; (semi-colon)
{SHIFT =}	SHIFT =

If you see:	Type:
{A}	CTRL A
A	INV. VIDEO A
{A}	INV. VIDEO CTRL A

BASIC XE™ Gives Your Atari 130XE™ All The Performance It Should Have Had In The First Place



Available July 1st

In the home computer races, the Atari 130XE stands out as a price leader. But using underpowered Atari BASIC™ on this otherwise fine machine is like racing in the Indy 500 with half your cylinders missing.

So don't get left at the starting line with only half an "engine." Change to the performance leader **now!** Buy BASIC XE from OSS, the **only** programming language designed especially for the Atari 130XE.

Just look at what you get for one low sticker price:

BEST MILEAGE: With over 60,000 **more** bytes for your programs, BASIC XE lets you use all the memory you paid for.*

MORE HORSEPOWER: Run Atari BASIC programs 2 to 6 times faster.* Even with its incredible power, BASIC XE is compatible with Atari BASIC.

BETTER HANDLING: With auto line numbering, renumbering, program cross referencing, English error messages, and more.

CLASSIC DESIGN: Show off the sleek **structured** style of your own programs when you use BASIC XE statements like PROCEDURE, IF...ELSE, and WHILE...ENDWHILE.

FREE ACCESSORIES: Get over $100 worth of Atari BASIC options FREE when you buy BASIC XE: complete Player/Missile Graphics support, string arrays, DOS access, SORT commands, readable listings...over 50 extras at no additional charge.

- If you're ready to step up to real performance... YOU need BASIC XE now!
- If you haven't written your first BASIC program... YOU need BASIC XE now!
- If you're already a real pro in BASIC... YOU need BASIC XE now!
- **BASIC XE may well be the best buy any Atari owner ever made.**

OSS™

*Want to know more? Call or write for free brochure or ask your local dealer.

Atari 130XE™ and Atari BASIC™ are U.S. registered trademarks of Atari Corporation.

Optimized Systems Software, Inc.
1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099

ERROR MESSAGES

Atari Never Told You About

[After the mass firings in Sunnyvale when Jack Tramiel took over, certain documents marked "Top Secret" were found in a trash can at a rest area on I-80 East. The MACE Journal is proud to be the first to present this valuable information to the world of Atari users. -Ed.]

Note: The following error messages result from external malfunctions, including operator error, and are only implemented on machines containing the experimental PSI (Pretty Small Integrated) chip. To determine if your machine has the PSI chip installed, set up a loop to read location 53770 ($D20A), the random number generator. Concentrate on a number between 0 and 255; if you can force the output of 53770 to equal your chosen number more than 87.45% of the time, you are one of the lucky few with the PSI chip.

ERROR 256 - Operator Negligence
You failed to stare intently at your 410 or 1010 recorder during the entire CLOAD process. Rewind the tape to within .01 mm of the original recording position, take a deep breath, and hold it while watching the tape grind through the recorder. It has been shown that blinking during a cassette load can set up shock waves sufficient to knock the tape head out of alignment and abort the load.

ERROR 257 - Keyboard Adhesion Error
There is peanut butter or some other sticky substance underneath your keycaps. (This error does not occur on Atari 400's with the original membrane keyboard.) You can try to pry off the keycaps yourself and clean up the mess, or face ridicule by bringing your machine to the service department of your local computer store.

ERROR 258 - Disk Damage Error
Your toddler has been trying to play your disks on his Fisher Price Record Player. This error can also occur when disks have been used as Frisbees, coasters, or to jimmy a lock.

ERROR 260 - TMF Error
A Transient Magnetic Field has erased all of the data on your disks and/or tapes. You are now the owner of 253 flat black plastic squares which can be used (with little success) to tile your rec room floor.

ERROR 261 - Release Date Shock
A previously-announced Atari product has been released on time; the shock was too much for your computer, which will be inoperative for the next three months, thereby bringing things back to normal.

ERROR 262 - Poor Programming Technique
You have aggravated your BASIC cartridge by writing "spaghetti code", full of tangled GOTO statements. In retaliation, it has renamed all of your variables as carriage returns (CHR$(155)). No recovery possible.

ERROR 263 - Late Night Error
This error most often occurs at 4 am and is due to the fact that the computer is being put to sleep by your yawns. Grinding No-Doze between the cartridge and its connectors will prevent this error, as it will totally disable your system and you will be able to get lots of sleep in the next few months while you are waiting for your machine to be repaired.

ERROR 264 - Fed Detected
An FBI informant has tapped into your phone line and is monitoring your pirate downloads. For $59.95 (plus $2.00 shipping and handling) you can buy an Honesty Chip which will immediately switch the transmission to a public domain program. The Honesty Chip is available from I. M. Cott, Cell Block 534, San Quentin, CA 94013.

ERROR 265 - CUI Error
The operator is guilty of Computing Under the Influence of proscribed substances. Power down and wait for operator detoxification before attempting further operation.

ERROR 266 - Malfunction Timeout
It has been too long since an error has occurred, so the Error Generator, ERRGEN, at location 49155 ($C003) has chosen to spice up your life. This error is seldom seen because of the unlikelihood of operating your computer for more than 15 minutes without an error.

TWO STRATEGIC GAMES

Reviewed by Charles M. Hostetler

[The reviewer reports that he has been a strategic games fan since the early 60's, when he started playing "Blitzkrieg" and "Battle of Britian". Since then, he and his wife Susan have collected some 2000 or so "simulations", plus a number of boardgames.

He purchased his Atari 800XL in the winter of '84 in hopes of using it in his law practice, but managed to write only one single-page will, which took about 8 hours, since he didn't have a word processor and had to use the "LPRINT" command. He has since been employed by an insurance company and can dedicate the time spent with his Atari to playing and reviewing games. He has been a member of MACE since last fall. -Ed.]

STARWARRIOR

Epyx

I picked up this disk for $8.00 at a closeout of Epyx software; I think it was well worth the money. Star Warrior comes with a rule book and two 4 x 8 cards: one for a summary of the 19 Commands (this is a keyboard, not a joystick, game), and one for a two sided map of the playing areas.

In Starwarrior you play a native of the Planet "Fornax" who, with others, has decided that the recent acquisition of your homeworld by the Stellar Union has led to unpleasant side effects, such as "taxation without representation". As such you hope that the acquisition of two "Furies" (suits a la Heinlein's Starship Troopers) can allow your group to "make a statement" and lead a revolt. The statement you hope to make is the destruction of the Military Governor.

That is Scenario Two in the game. Scenario One is the diversion of Union forces whilst your compatriot does in the nasty Governor. Your mission #1 is limited by time. You can choose your own time limit, which sets the time before "RECALL" when you are to return to your starting "square" to be successful. In mission #2 there is no set time limit, but the longer you spend trying to blow up the Governor, the more energy you use.

There are nine types of enemy units varying from infantry which can be ignored as long as your shields are up on maximum, to heavy tanks and Nitron Guns (the latter being a type of artillery best shot at before being shot by). You have a choice of three basic suits: the Dragoon, Marauder, and Ninja; or you can design your own from components. (There is a cost chart).

To get around, you can "fly", "jump", or "walk". Flight is recommended for rough terrain. When flying it is a good idea to zigzag as straight flight does seem to attract more enemy units than needed.

You are not without weapons. Your Powergun is effective at close range provided its range is properly set. I wouldn't recommend it too highly, however, as it's not as sure as a well-placed missile, or the Blaster. Also, the Powergun uses the same energy that the rest of the systems use, including your shields and flight power. The rulebook calls the Powergun the weapon of choice for the Ninja suit as it is the quietest and least likely to attract other "baddies".

The blaster fires three charges that have a cumulative effect (unlike the Powergun). Naturally, the number of charges you have is limited.

Missiles explode in a mushroom cloud and are targeted by direction and range. These too are limited in number. They are also the only way to destroy "installations", which are worth more points than units and are where the Governor can be found.

"Installations" are of two basic types: Towns and Forts. These may be civilian or military. The only way to tell is through your "O"bserve command. This is particularly important as points are deducted for doing in innocent civilians.

The screen display is about 80% a square that represents a terrain type. These are mountain (good graphic background), clear, swamp, and forest (all of average graphics). Rather than scrolling, exiting a square blanks and flashes the screen to the next square. You can move diagonally. The entire playing area is about 80 squares. These repeat the last terrain rather than wrap around to the "other" side. I've never visited half a field in dozens of games.

You have available an "invisibility" command with the Ninja suit and a Decoy option with the Marauder. Both of these options are helpful when your shields are down and your medical system is failing. (Your wounds are regularly healed at a modest rate, otherwise.)

You also have about 12 "windows" on your display that tell your direction, number of missiles, charges for your blaster, energy units left, the type of enemy unit present, if any, etc.

I thoroughly enjoyed this game, and if you like strategic simulation games, you probably will, too!

SARATOGA

APX Classics from Antic
Program by Paul Wehner

This is a game simulation of 1777, when the British had their last, best chance to win the American Revolutionary War. The British plan was to concentrate their three armies, and in the process, eliminate Washington's Continental Army as an effective force. As history tells us, Washington met and defeated Burgoyne (Gentleman John) near Saratoga.

The game uses the system of "Eastern Front", allowing the solitary player to use the joystick to control his American forces and read the British dispositions. The cursor also scrolls the screen through the 15 screen map. Forests, mountains, cities, rivers, and forts are depicted with good graphics. Unfortunately, the one lack of the system is the lack of place names. (This may be an inherent limitation of the system.)

Battles are joined by commanding your unit to a square occupied by an enemy unit. When all your orders are in and START is pushed the battles are fought, complete with sound effects. The usual way to destroy enemy units is through surrounding them, although repeated attacks can destroy a unit. Retreats are handled automatically by the computer.

The documentation is all on the reverse side of the disk, a feature that I wish more companies would consider. The documentation can be printed or read on screen. There are multiple scenarios from which to choose, including one simple enough for beginners to learn the system and still win. Victory is determined by control of such places as New York, Boston, and West Point (Washington, D.C. not being around yet). Supplies and seasonal changes (the Revolutionary forces double in the summer) are handled automatically, also. Overall, this is an excellent game and well worth the $15.95 price tag from Antic Arcade.

Innovision

Touch is a two to fifty player adult party game. Players take turns 'touching' other players.

"Touch, the adult party game by Innovision, is really unique! ... maybe a 'very liberal, open minded, and willing to share' adult game is more like it."
-Kirk Revitzer
MACE Journal, June '85

Vigitizer is actually a voice digitizer that plugs into any joystick port. This "electronic tape-recorder" allows you to record your voice, save it to disk, and play it back. A special feature allows you to play your voice backwards. The Vigitizer includes BASIC and Assembler load and playback routines for use in your own programs.

Plexus is the first micro-computer bulletin board to understand plain English commands. Commands like "read new" to read new items, and "help editor" to get help for the on-line text editor. All DOS functions are supported remotely for privileged users. Plexus is written entirely in C and machine language.

Make check or money order payable to: Innovision

Mail to: Innovision
P.O. Box 35
Milford, MI
48042-0035

PRICES
Touch ----> $10.00 + $2.00 S&H
Vigitizer > $30.00 + $3.00 S&H
Plexus ---> $50.00 + $3.00 S&H

EDUCATIONAL SOFTWARE

by Hugh McLean
Co-Sysop, BUNKY'S BOARD (313) 546-3689

My interest in educational software is fairly deep-seated, and while neither extensive nor exhaustive, it is intensive. One of my first real computer projects was a rather lengthy educational program for BUNKY, which proved to be beneficial for both of us. I am somewhat disdainful of the psychological advertisements that subtly suggest my parental inadequacy and primordial myopia because I don't rush to the local computer store and purchase a computer for my college bound child. If I would just be responsible as a caring parent should, my child would have a computer and be a successful student. The rather ambivalent equate that "student failure" is equivalent to "parental frugality" irritates me.

Another thing that perturbs me is the hype of availability of "educational software", that ubiquitous term which delineates inferior systems by virtue of volume of programs which are designated as "educational". Many of these "educational" programs advocate the advantage of "hand/eye coordination", and after all, we certainly wouldn't want our children to be deprived of that. Hand/eye coordination, in my opinion, is that inane ability to generate a higher score in PAC-MAN or DONKEY-KONG, and has absolutely nothing in common with S.A.T. or A.C.T. scores.

I was excited about purchasing a VCR a few years ago - excited because at last I could rent some really good tapes and enlarge the horizons of my educational pursuits by studying some of the topics I was too busy to pursue formally. Alas, my enlightenment was only an ambitious dream, for I found almost nothing available. Sick, trashy, x-rated movies abound, but try to find something on archaeology, or even the French language. With current technology making color-sensitive, erasable, optical disks coupled with computer controlled laser machines that can function frame by frame, or sequentially, an educational tool exists that is totally amazing and virtually unexplored.

Today's computer technology reminds me of the early 1900's when automobiles were invented before the support systems of macadam roads, expressways, gas stations, lead free fuels, synthetic elastomers, curved shatterproof wind-shields, and AM/FM stereo tape players.

I anticipate that personal computers are more than a passing Hoola Hoop fad, and that they will be the progenitor of new tools that will provide as yet unimaginable conveniences tomorrow that will be as common as gas stations are today.

I recently read an article in the March '85 High Technology magazine entitled "The future of Educational Computing", by Moses T.L. Ma that was as rare as it was refreshing. According to the article, learning has four levels, each one associated with a different type of mental operation:

1. MEMORIZATION: This is usually a repetitive operation where a student learns by rote and doesn't understand what he is memorizing, or its application: math functions, spelling, color and symbol associations, etc. This is the prime area that educational software uses, since repetitive functions make good use of the computer.

2. UNDERSTANDING: This level of learning is the process by which underlying principles are discovered. The student finally realizes the function of his memorization efforts.

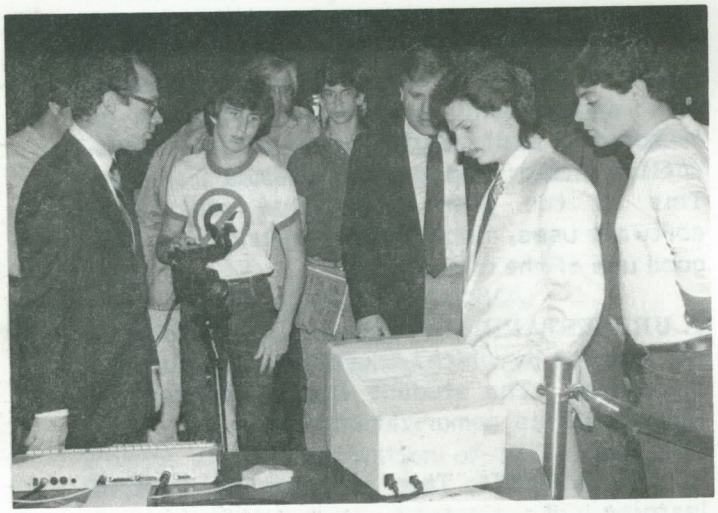
3. CREATIVITY: This progressive level of learning is achieved when the student develops his ability to think for himself, using the understanding that he has acquired. Educational software hasn't really done much in this area, understandably, since it is easier and less time-consuming to develop "memorization" software.

4. WISDOM: "This level goes beyond memorizing efficiently, understanding completely, and creating beautifully, to the ability to see underlying realities."

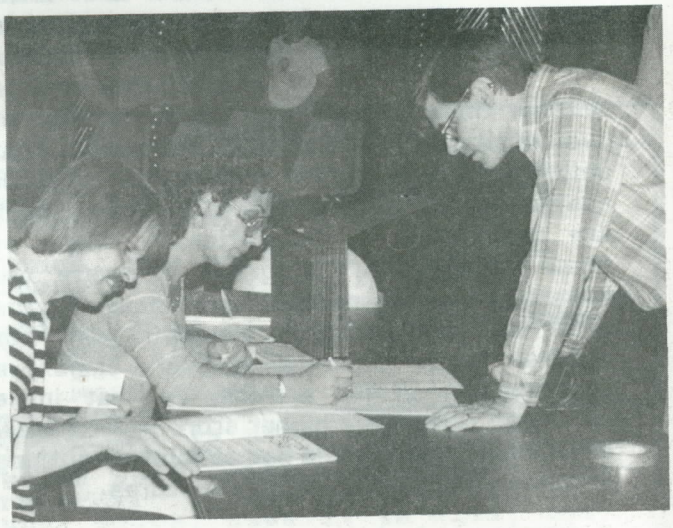
While these four levels of learning may seem a bit beyond the scope of the personal computer, they nevertheless represent goals to which educational software should aspire.

Jan Landis displays our Birthday Cakes

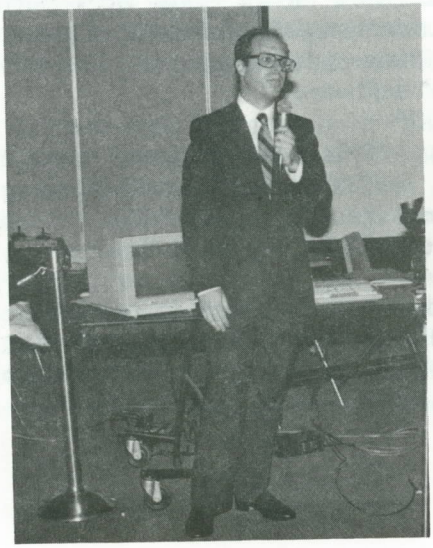
Kirk Revitzer and Dave Duberman cut our Birthday cake

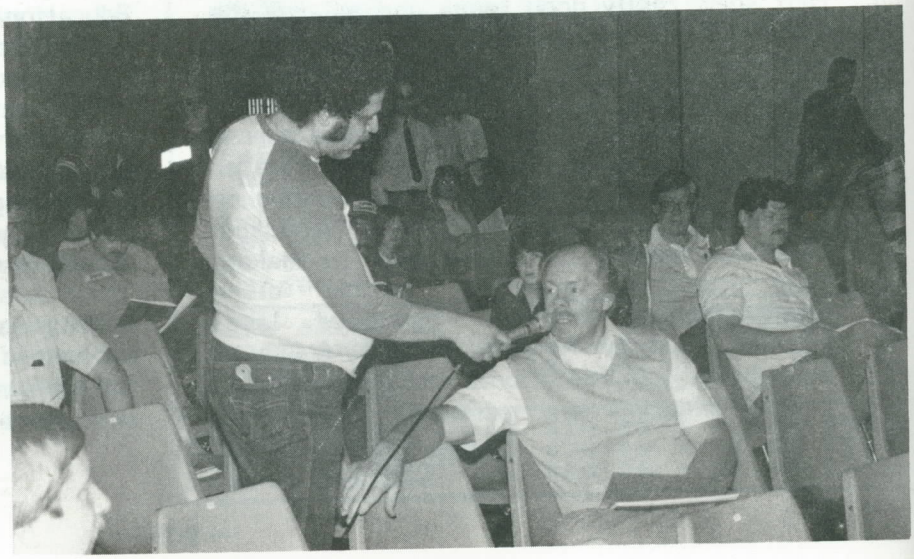


Dino Roggero raises important questions about the 520 ST.

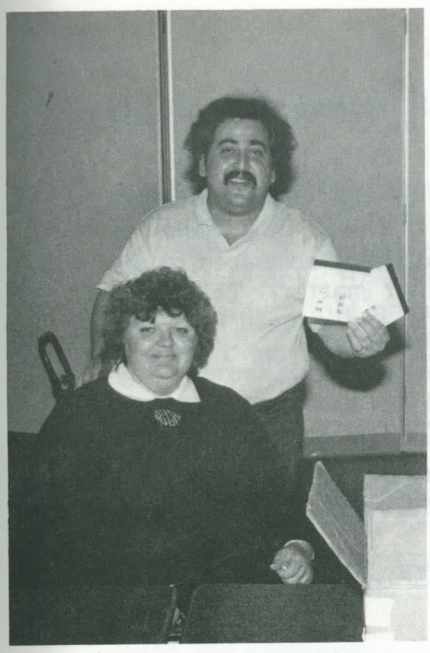


Sharie Middlebrook and Burt Gregory tally renewal memberships



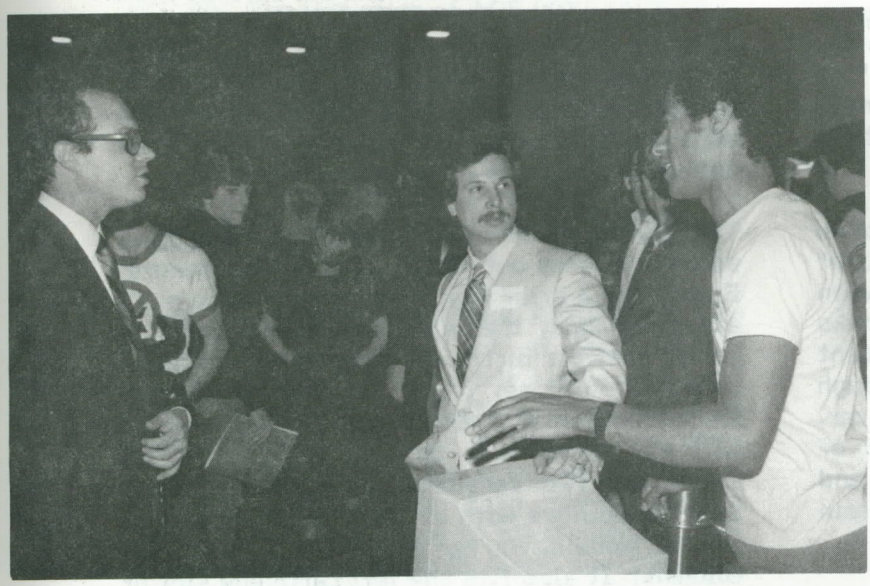


Dave Duberman our guest speaker answers questions from MACE members



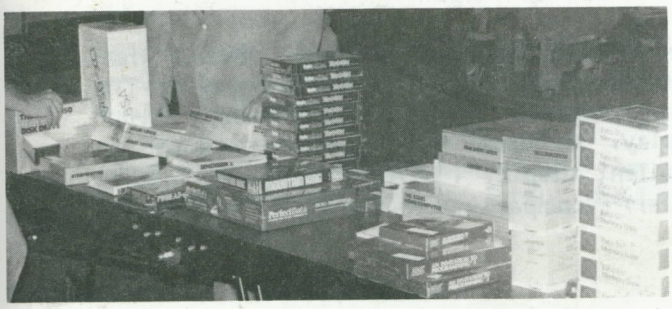
Dave & Bev Zappa custodians of our Disk Library

A full house for our 5th Birthday celebration



Dave Duberman explains the 520 ST to Dino Roggero and Alva Thomas

Dave Duberman gives an exclusive demonstration to Eric Wujicik (Detroit News) and Howard Kenig (WHYT)



Some of the items given away to the MACE members

RALLY SPEEDWAY
Adventure International

Reviewed by Jay Slatkin

Did you ever run across a game that wasn't long, wasn't complex, but was just plain fun? Well, those games are hard to come by these days, but I've just run across one here.

The object of RALLY SPEEDWAY is simple: get across the finish line alive before your opponent does. When you first boot up the game you get the main menu for a brief moment and then the title screen. It's not very "classy", but the game play more than makes up for it.

You have several options to choose from, such as the type of track. You have your choice of a dry, wet, or icy track. I usually choose wet because the game is more fun on a slippery track but in my opinion an icy track is too slippery. You have a choice of your "top speed". You can choose 40, 80, or 100 mph. You also have a choice of rate of acceleration: fast, medium, or slow.

You also have another option which is nice to have. The two choices are called "Real Life" or "Only in a Computer". In "Real Life" when you're speeding around the track and make contact with a tree or house you have a violent crash and you see your driver rolling through the flames as he's tossed out of the car. "Only in a Computer" is nice. Instead of crashing, you drive through the houses and trees! You can cut across the wooded terrain and onto the track again. I think it's fun.

In addition, you can not only make your own track, you can save it to disk! When you create a track you can start from scratch, or alter any of the two tracks programmed into the game and save those to disk as well.

Playing with two players is a different story altogether. You can take out your aggressions on your opponent by bumping him into a violent crash. When you get so far ahead of him that your opponent is going off the screen, he obtains a 5 second penalty. I found these quite frequent in game play. Even the most masterful drivers will receive this penalty often.

Comparing this game to the amount of disk space it takes up, I think this game is excellent! I find it excellent compared to any car-driver game for that matter. This game contains brilliant sound and smooth four-way screen scrolling. On a scale from 1 to 10 I give it a very solid 9.

ACE 80 CARTRIDGE
Magic Software

Reviewed by Murray D. Kucherawy

I love my Atari 800. I bought it in 1982 and when I got tired of playing games, I installed a BIT-3 board, purchased the 80 column version of Letter Perfect and did serious word processing on a letter quality printer. My wife edits a medical journal and after seeing my system, she purchased two for her office. Her office is in a university hospital and she still chuckles over the raised eyebrows when she talks about her Atari word processor.

The 800XL with its "closed" construction seemed to shut out third party entrepreneurs. Except for a brief flurry concerning Atari's XL expansion box, the 80 column capability was gone. MACE's birthday party, however, gave us a real surprise. Magic Software of Monroe, MI introduced ACE-80, a 12K cartridge which will put 80 columns on your TV or monitor. There are two versions. 80XL works in all Atari computers with at least 48K. In the XL and XE models, you can program in BASIC in 80 columns. It also fits the left cartridge slot of the 800 but, of course, you can't simultaneously use BASIC. ACE-80 is a right cartridge slot version (yes, you heard correctly!) which is specifically designed for the 800 so that the left slot is available for your BASIC cartridge.

The central question of this review then becomes "How legible is the display?". The answer depends on what you are using for your display. A monochrome monitor produces an excellent display ("WOW!", to quote a local dealer) - comparable to the BIT-3 display. A

color monitor takes away some of the resolution but it is still good. A color TV gives a good/fair display depending on how well the set has been tuned but I could not recommend it for extended use. I was unable to test it on a B/W TV but my guess is that the display would be very good on a properly tuned set. I recommend you see the display on your terminal before you purchase. The final screen appearance can be "fine-tuned" to your liking by use of the START and SELECT keys. Unfortunately, this adjustment must be performed at each session. My suggestion to the designers would be to have a file written to disk after initial set-up so that subsequent screen set-ups would be automatic.

With the cooperation of LJK, Magic Software sells reasonably priced versions of LETTER PERFECT, DATA PERFECT and SPELL PERFECT which have been modified to run with ACE 80. The list price of ACE 80XL/ACE 80 is $49.95 . At less than 25% the cost of an 80 column board, this is a very attractive way of upgrading your Atari to professional levels.

GREETINGS FROM YOUR EDITOR...

First, I would like to thank EVERYONE who pitched in and helped out at last month's meeting. When we found out just a few days before the meeting that Dave Duberman from Atari actually would be attending, we knew that attendance would be up and we would need even more volunteers than normal for a Birthday meetings. Due to all the help we got from our members, things went smoothly and everyone seemed to have a good time.

Thanks are also in order for our advertisers and supporters who donated so many door prizes. Remember these merchants when it's time to go computer shopping! Sector One International, RiteWay, Family Computer Center, Just Software, Software Library, OSS, Sy Draft, and DataWorld Distributing. (Hope I haven't left anyone out!)

Keep those submissions coming in - we have had some very good articles lately, and I hope they continue.

SY DRAFT
OFFICE, ART & DRAFTING SUPPLIES

YOUR WORD PROCESSING CENTER

SPECIAL PRICES FOR MACE MEMBERS

SSDD DISKS:	10 for	$ 6.98
	100 for	$ 65.00
	1000 for	$600.00
TIGER HATS		$ 1.99
DISK PUNCH		$ 5.00
DOUBLE-UP DISK PUNCH		$ 8.95
INDUS GT DRIVE		$227.00
MPP 1000E MODEM		$ 96.00
MPP MICRO STUFFER		$103.00
ATARI PRINT SHOP		$ 30.00
KOALA PAD - ATARI		$ 29.95
FLIGHT SIMULATOR		$ 34.00
BMC PRINTER BX80		$185.00
SCM D100 PRINTER		$190.00
PANASONIC 1091 PRINTER		$279.00
DATA CASE (HOLDS 50)		$ 8.50
MEDIA MATE (HOLDS 50)		$ 9.50
DISK CASE (100 W/LOCK)		$ 13.00

WE HAVE HARD-TO-FIND ITEMS LIKE:
COLOR PRINT SHOP PAPER
PRINT SHOP ENVELOPES
COLOR PRINTER RIBBONS
SOFTWARE IS ALWAYS 30% OFF

THE SHELL GAME

CRACKING ATARI LOGO

by Ann McBain Ezzell

Working with Logo and turtle graphics can help children (and adults) develop the ability to estimate angles and distances. This month's "game" will offer practice at several different levels as the user tries to follow the path drawn by a turtle. This game could be improved by the addition of sound effects and/or color changes; feel free to tinker with it as you wish. If you are feeling adventurous, you might think about adding some error checking to make sure that the leader turtle doesn't go off the edge of the screen.

To play the game, type in the procedures and global variables as listed here, then type "FOLLOWME". You will have to choose a level of play from 1 to 4. On all levels, the leader turtle draws a path by choosing an angle and a distance and adding them to a list of commands. The possibilities increase with increasing levels of difficulty. Level 1 allows only RT 90 and LT 90, and the distance is always FD 20. Level 2 adds in RT 45, LT 45 and FD 40. At Level 3, the turtle can also turn RT 30 or LT 30, and go FD or BK 20, 40 or 10. Level 4 adds a 60 degree turn and a distance of 30 (FD or BK). At Level 1, you must match 5 moves to complete the game successfully; this number increases by one for each higher level.

The main procedure calls procedures which introduce the game, get the level of play, run the game, and ask if you want to play again.

```
TO FOLLOWME
TITLE
TYPE [ENTER LEVEL OF PLAY ( 1 - 4 ) :]
GETLEVEL
CS
TELL [0 1 2 3] HT
MAKE "PATH []
RESET
PLAY
FINISHED
END
```

TITLE prints out an introduction to the game and calls the procedure WAITKEY with an input of 155, which is the ATASCII code for a carriage return. This will make the program loop until the <RETURN> key is pressed. WAITKEY can be used whenever you want to wait for a specific keypress.

```
TO TITLE
TS CT
PR "
PR [FOLLOW THE TURTLE...]
PR "
PR [WATCH THE TURTLE AS IT ADDS
STEPS]
PR [TO ITS PATH AND TYPE IN THE
PROPER]
PR [COMMANDS TO MAKE YOUR TURTLE
FOLLOW]
PR [THE SAME PATH.]
PR "
PR [HIT < RETURN > TO CONTINUE]
PR "
WAITKEY 155
END

TO WAITKEY :N
IF EQUALP RC CHAR :N [STOP]
WAITKEY :N
END
```

The next step is to get the difficulty level for the game. The prompt is printed in TITLE rather than GETLEVEL so that the program can loop through GETLEVEL until a valid response is given (1-4) without printing out the prompt each time. If the input is a number and if it is between 0 and 5, the number is echoed back to the screen and the program waits for a carriage return to continue the game. This WAITKEY 155 could be eliminated, but many people are used to pressing <RETURN> after an entry. An extraneous <RETURN> can often mess up your program, so it's a good idea to plan for one.

```
TO GETLEVEL
MAKE "LEVEL RC
IF NUMBERP :LEVEL [IF AND :LEVEL > 0
:LEVEL < 5 [PR :LEVEL WAITKEY 155 STOP]]
GETLEVEL
END
```

RESET clears the screen before the start of the game and after each successful round during the game. It also puts the turtles at

position [0 20] to put them nearer to the center of the split graphics screen so they are less likely to run off the edge. When moving the turtles, the pens must be up so that they won't leave a trail back to the center.

```
TO RESET
CS
TELL [0 1] HT PU
SETPOS [0 20]
SETH 0
PD ST
END
```

PLAY is heart of this game; it controls the leader and follower turtles and checks to see that the two paths are the same. First it calls MAKEPATH to start the selection of the leader turtle's path. MAKEPATH uses a procedure ITEM which is from the Atari Logo Reference Manual, page 59. ITEM will return a specified item from within an object, whether word or list. There are four lists set up as global variables which contain the choices available to the leader turtle. There are only two choices for turning, RT and LT, but the other lists each contain four choices for the path. ("DIR has two FD's and two BK's because Levels 1 and 2 only allow forward motion, while 3 and 4 allow either direction.) Depending on the value of :LEVEL, a random choice will be made from each of the lists in turn and the commands will be added to the list :PATH. (The higher the level, the greater the possible value for the random number.) Each time MAKEPATH is called, :PATH will be lengthened by four items (for example, RT 90 FD 20).

PLAY then tells turtle 1 to execute the commands found in :PATH. You must type in commands to make turtle 0 follow the same path. If the two lists (:PATH from MAKEPATH and :PATH2 as read from the keyboard) are not equal, PLAY calls MISTAKE, which prints out the correct list of commands and the number of steps matched correctly. AGAIN asks if you want to try again, returning control to TOPLEVEL if the answer is other than "Y".

If the two paths are the same, and the game has not reached the maximum number of rounds for that level, PLAY calls itself after executing a RESET and a DELAY. DELAY is simply a recursive procedure which counts down from its input to 0.

```
TO PLAY
MAKEPATH
TELL 1 SETPN 0
RUN :PATH
CT PR [ENTER THE COMMANDS:]
MAKE "PATH2 RL
TELL 0 SETPN 2
RUN :PATH2
IF NOT EQUALP :PATH :PATH2 [MISTAKE]
IF 4 * ( :LEVEL + 4 ) > COUNT :PATH
[DELAY 20 RESET PLAY]
END

TO MAKEPATH
MAKE "PATH LPUT ITEM ( 1 + RANDOM 2 )
:TRN :PATH
MAKE "PATH LPUT ITEM ( 1 + RANDOM
:LEVEL ) :DEG :PATH
MAKE "PATH LPUT ITEM ( 1 + RANDOM
:LEVEL ) :DIR :PATH
MAKE "PATH LPUT ITEM ( 1 + RANDOM
:LEVEL ) :DIS :PATH
END

TO ITEM :N :OBJECT
IF :N = 1 [OUTPUT FIRST :OBJECT]
OUTPUT ITEM :N - 1 BF :OBJECT
END

MAKE "TRN [RT LT]
MAKE "DEG [90 45 30 60]
MAKE "DIR [FD FD BK BK]
MAKE "DIS [20 40 10 30]

TO MISTAKE
CT PR [SORRY... WRONG MOVE]
PR [THE MOVES SHOULD HAVE BEEN:]
PR :PATH
TYPE [YOU MATCHED \ ]
TYPE ( COUNT :PATH ) / 4 - 1
TYPE CHAR 32
PR [MOVES]
AGAIN
END

TO DELAY :N
IF :N = 0 [STOP]
DELAY :N - 1
END
```

If PLAY cycles through the maximum number of rounds without an error, control passes to FINISHED, which offers congratulations and calls AGAIN to ask if you want another try.

```
TO FINISHED
CT PR [CONGRATULATIONS!]
PR [YOU FOLLOWED THE TURTLE!]
AGAIN
END

TO AGAIN
TYPE [DO YOU WANT TO PLAY AGAIN?]
MAKE "ANSWER RC
IF EQUALP :ANSWER "Y [FOLLOWME] [CT
,CALL 39929]
END
```

This game is not fully polished; it would be easy to add color and sound to liven it up. Less simple would be adding error checking (which is why I didn't do it). You might change the flow of the program so that MAKEPATH is called the necessary number of times for each level to create the entire path before drawing any of it. That way you could check to see that the leader turtle did not go off the edge of the screen. Then you would have to write a procedure to check the input list against the appropriate part of the leader's path. There may be an easier way to control the turtle; I'd be glad to hear it.

BURNOUT ALERT!!

Your dedicated Editor is running out of ideas for this column. Maybe you are getting tired of reading it - if you're not, SOMEBODY had better come out of his or her shell and send in some suggestions. Better yet, send in some of your own procedures or Logo activities. You don't have to have a perfectly complete program; half of the fun of Logo is tinkering with it and tailoring it to fit your needs. Maybe someone else has the perfect solution to that problem that has been bugging you for months. You'll never know unless you ask.

NEW USERS FORUM

by Tom Sturza

You say you can't tell a bit from a byte? You finally purchased a disk drive to replace that slow cassette, but DOS is a foreign language? AtariWriter is great, but why won't it underline? Your family bought a modem, but bulletin boards "hang up" on you when you call! IS THAT YOUR PROBLEM BUNKY?

Well, MACE has just the thing for you and your family - the New Users Forum. We'll do our best to answer your questions, show you how to use your Atari for something besides games and explain "How things work!"

We'll meet monthly at the Southfield Civic Center, in one of the upstairs meeting rooms at Parks & Recreation (across the hall from the Pavilion). Meetings will be on the fourth Monday of each month. The first three meetings will be on June 24th, July 22nd and August 26th. These meetings are free; however, seating is limited, and MACE membership cards will be required for admission.

So, if you are interested in attending the New Users Forum, get those questions and/or requests ready. Please call me (from 7-10 pm at (313) 477-2345) if you would like to have a specific topic or software package discussed at the Forum and I'll try to have an "expert" there to speak on the topic.

MACE SIGS

The following Special Interest Groups are still active. Contact the person listed for more information.

Atarimusic SIG: Mike Lechkun, (313) 978-8432 or MACE EAST BBS, (313) 978-1685

FORTH Interest Group: Tom Chrapkiewicz, (313) 562-8506 or 845-4570 x60

SIG-ED: Mark Kennedy, (313) 465-5849 evenings

SECTOR - ONE

INTERNATIONAL

I ♥ MY ATARI

POWER WITHOUT THE PRICE

SECTOR-ONE INTERNATIONAL

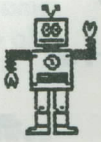
COMPUTER MAIL ORDER WORLDWIDE

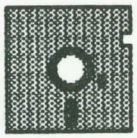
37220 TRICIA DRIVE — STERLING HEIGHTS, MI. 48077

ORDERS: 313-978-2208 8AM-8PM — MODEM: 313-978-8087 24 HOURS

SEE US AT M.A.C.E.

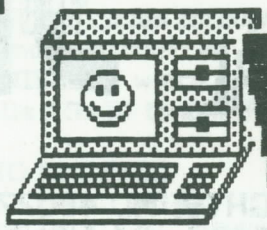
130XE 128K	$159.00
800XL 64K	$ 99.00
850 INTERFACE	$114.00
1200XL 64K	$ 59.00
1010 RECORDER	$ 39.00
1020 PRINTER	$ 49.00
1025 PRINTER	$179.00
1027 PRINTER	$249.00
1030 MODEM	$ 69.00
1050 DISK DR.	$169.00
TOUCH TABLET	$ 44.00
CX85 KEYPAD	$ 29.00
HAYES 300	$149.00
HAYES 1200	$399.00
MOON PATROL	$ 16.00
SKYWRITER	$ 16.00
TRACK & FIELD	$ 21.00
PRINT SHOP	$ 35.00
SYN CALC	$ 39.00
SYN CHRON	$ 34.00
SYN COMM	$ 34.00
SYN FILE	$ 39.00
SYN STOCK	$ 34.00
SYN TREND	$ 34.00
S.A.M.	$ 44.00



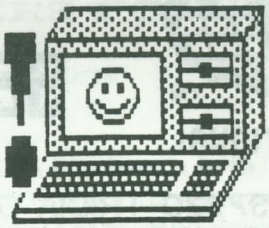


6 FT. I/O CABLE	$12.00
I/O CONNECTORS	$ 2.50
850-HAYES CABL	$19.00
850-EPSON CBL	$24.00
MPP 1000-E	$99.00
MPP 1150 I.F.	$59.00
MPP 1152 BUFF	$14.00
MPP 1156 S.I.F.	$40.00
MPP MIC. PRT.	$49.00
JOYSTICK CABLE	$ 2.00
VG170 JOYSTICK	$ 5.50
JOYSTICK 2 BUT	$ 6.50
VG172 PADDLES	$12.00
MONITOR CABLE	$ 4.00
1020 PEN PACK	$ 2.50
1020 PAPER PK.	$ 3.90
1025 RIBBON	$ 1.75
1027 INK ROLLER	$ 3.50
820 RIBBON	$ 2.00
825 RIBBON	$ 2.75
GEMINI RIBBON	$ 1.75
SP-80 RIBBON	$ 8.00
EPSON RIBBON	$ 4.50
C. ITOH RIBBON	$ 5.00
CALL FOR MORE INFO	

SHIPPING: 5% AIR, APO, FPO: 8% — COD WELCOME IN USA: ADD $2.00



WARNING !



This machine is subject to breakdowns during periods of critical need.

A special circuit in the machine called a 'critical detector' senses the operator's emotional state in terms of how desperate he or she is to use the machine. The 'critical detector' then creates a malfunction proportional to the desperation of the operator. Threatening the machine with violence only aggravates the situation. Likewise, attempts to use another machine may cause it to also malfunction. They belong to the same union. Keep cool and say nice things to the machine. Nothing else seems to work.

Never let anything mechanical know you are in a hurry.

Designed and Printed by Phyllis Evans
Using PRINT SHOP and HOMETEXT

A TECHNICAL VIEW OF THE CES

by Todd Meitzner

My primary reason for going to the Consumer Electronics Show (CES) was to see what was in store for Atari. Although Atari didn't seem to offer any new hardware additions to the 8-bit computers, others did. ICD was there with their "US Doubler" for the 1050 disk drive, which allows you to use true double density on your 1050. It comes with their SpartaDOS Construction Set. Also shown by them was their clock card which plugs into the cartridge slot (it still allows other cartridges to be plugged in as it includes a cartridge jack at its far end). It is supported by their SpartaDOS for time dating of disk files.

Although it is not out on the market yet, BATTERIES INCLUDED had their 80 column board working on their Atari. The display was shown on a green screen monitor. It plugged into the cartridge port and was about the size of a normal cartridge. I couldn't see if another cartridge could be plugged into it. The only bad part is that it will work on only the XL and 130 XE series computers (it seems there are some timing differences from the old 800/400s).

MPP was there sharing a booth. They showed off their line of products. While the literature on their hard disk interface hadn't arrived, we were able to get some information out of them. The entire package included the interface to the computer (an XL or 130XE), a SASSY interface to the hard disk, and the 10 megabyte hard disk. The price for this was under $1000. The DOS they preferred for the hard disk was OSA+ 4.0. They also indicated that the interfaces may be available separatly.

Telsys Computer Peripheral Products were there showing their printer interfaces. The one we saw also was able to dump the control characters in both their graphics form or in another form (e.g. ^A = Control-A). The price on the literature was below $60.00.

Although they were not there (I didn't see them) some literature was left in Atari's room from Quantum Microsystems, Inc. about their Q-Modem. It's a direct connect modem for the 8-bit Atari computers. It connects through the serial port and is also powered by it. It comes with its own software and is capable of auto-dial and auto-answer. They even claim to have a 1200 bps option board for it.

Unlike the 8-bit computers Atari did have some new hardware for the 16-bit ST computers. Atari in addition to showing the 520ST also showed the new 260ST.

The 520ST will be the computer store version and will most likely retain the operating system totally in RAM. It has to use either an analog RGB monitor or Atari's high-resolution monochrome monitor and has to use an external 1 or 2 sided disk drive (2 sided available later).

The new 260ST is the mass market version and has the operating system in ROM. Unlike the 520ST it has a TV interface but is also able to use the monitors. It also has a built-in single sided disk drive. The power supply is also built in. It will be available sometime after the 520ST is introduced.

Atari also mentioned that the 15 megabyte hard disk would be available for the ST sometime late this year.

As to other companies supporting the ST series, there were several. Haba Systems, a maker of software for the Macintosh, intends to port all its present and future products over to the ST. They make among other things word processors, file managers, electronic speedsheets, and communications software. They are also coming out with both 10 and 20 megabyte hard disks and a 1200 bps modem.

Also in Atari's room an independent company showed off the Compact Disk (CD) interface. They had on the CD an encyclopedia which they could access through the ST. Among other things it was capable of cross referencing an entry through many different subject headings.

A number of companies were coming out with software on the ST, although unfortunatly others took a wait and see attitude. Among the former are Batteries Included, which will

write in GEM for both the ST and IBM, Sierra On-Line and SubLogic which will be coming out with some games for the ST, Infocom (yes, Zork will be available on the ST), and others.

Other details of the ST were also talked about in the Atari room. Although Atari doesn't say they will expand the memory of the ST computers, they say there is nothing in the operating system to prevent this from being done.

Initial systems will come with Logo. While it's good at graphics and recursion, don't attempt to write a terminal program in it.

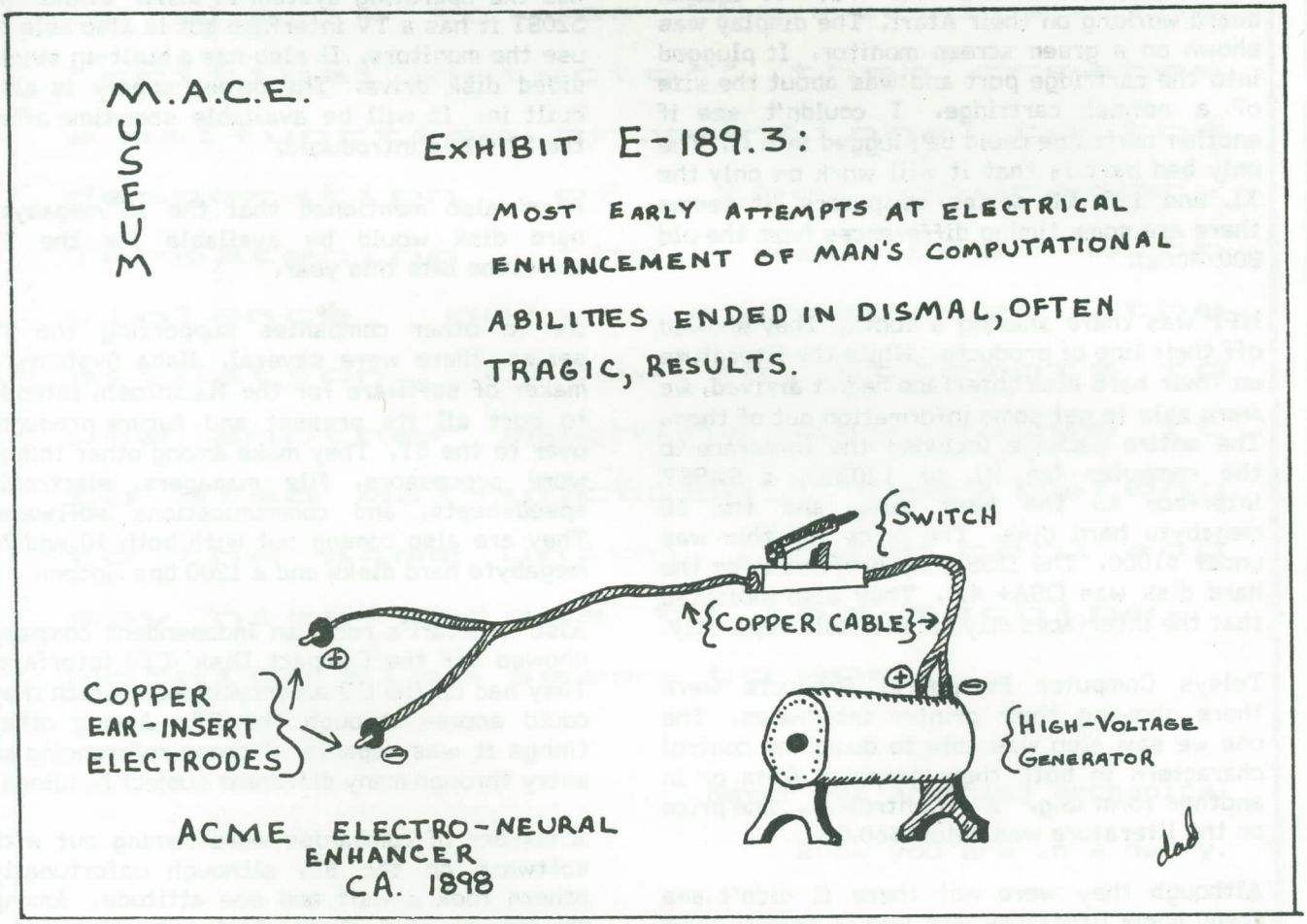
Forth was there and was booted up in a demo mode. It seems to be a 32 bit version of Forth and of course supports the grahics. They still didn't know if it would be released, though.

An alpha version of GEM-write was on a disk, however I didn't see it up and running.

Someone said that another version of "C" was just ported over from the Macintosh and was much easier to use.

No word yet if the development package would be available to the ordinary user. It was said that the information was too technical and that it may be rewritten for the ordinary user.

It appears that the ST can be hooked into a network. This can be done through one of three ways: through the serial port (9600 bps), through the MIDI interface (about 70k bps), or through the hard disk interface (1.33m bps). The last is by far the fastest and looks to be promising.



COMING SOON FROM CES!

PRINTER DRIVER CONSTRUCTION SET

THE ADVENTURE CONSTRUCTION SET

MINDWHEEL

ESSEX

BLUE MAX 2001

PAPER CLIP

KENNEDY APPROACH

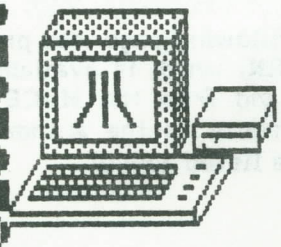
TURBOPRINT GTA
(INTERFACE TO PRINT ATARI GRAPHICS)

GUNSHIP

ACROJET

SILENT SERVICE

130 XE'S IN STOCK NOW
$169.95 INCLUDES SUPPORT SERVICE



FAMILY COMPUTER CENTER

YOUR ATARI CONNECTION
FOR ALL YOUR COMPUTER NEEDS
BERKLEY, MICHIGAN * (313) 543-0520

MAGIC LANTERN

by Richard Q. Fox [72356,242]
Lee C. Zion [70155,656]
Bob Geayer [71425,463]
Doug Lange [74365,610]
& Tom McComb [72456,1042]

48K and BASIC required, printer optional.

Reviewed by Ann McBain Ezzell

There are a lot of public domain programs floating around on local and national bulletin boards; some of them are obviously amateur and some are of professional quality. MAGIC LANTERN, in spite of a few problems with error handling, comes closer to the latter category than the former. It does what it sets out to do, and it does it well. What more can you ask of a program?

This program serves two purposes: it allows you to set up a "slideshow" of pictures created with the Atari Touch Tablet or Koala Pad, and it also lets you dump your pictures to a printer. The screen dump works just fine on my Gemini-10 printer; the control codes to set up the printer are written in such a way that they could easily be tailored to fit your machine.

MAGIC LANTERN is easy to use: after you see a title screen, you choose a disk drive from 1-4 for loading the pictures. The program will only accept numeric input in the proper range, but does not have any provisions for attempts to read from a drive which is not available. For example, if you respond "2" but only have one drive turned on, the program will end with an error. This is one of the little glitches which could and should have been smoothed out.

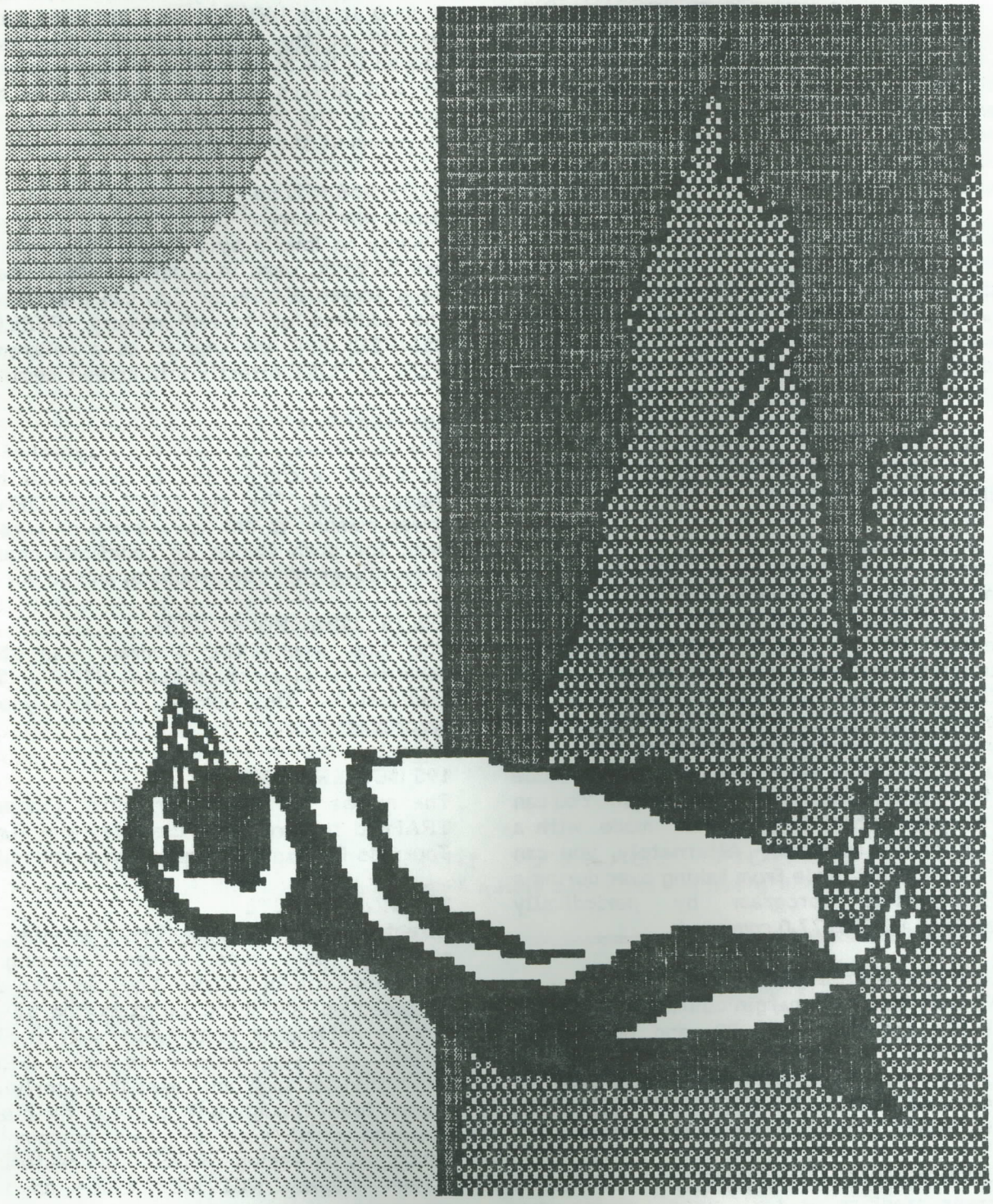
After you choose a drive, MAGIC LANTERN will read the disk directory and list all files having a ".PIC" extension. You can hit <ESC> to view all of the files, or use the cursor and <RETURN> keys to select certain ones. Pushing the space bar starts the show. While one picture is one the screen, the next is being loaded into an alternate screen memory area. The first picture "fades" (rather abruptly, but it fades) into the second one, then a third picture is loaded into the first screen memory area. This display flipping means that you don't have to sit and watch the picture being loaded in like you do with MicroIllustrator (the software which comes with the Touch Tablet and Koala Pad).

While the program is cycling through the files on the disk, you can use the <SELECT> and <OPTION> keys to pause and restart the show, or you can return to the main menu by pressing <START>. This will let you switch to a different picture disk. Switching disks without returning to the menu, will cause the program to look for the next file on the previous disk and halt with an error message.

You can also print the current picture by pressing "P". There is an error trap here, but it doesn't tell you what is causing the error. When the print routine encounters an error, the program simply returns to the main cycle and loads in the next picture, leaving the user to figure out what is going wrong.

I can't fault the printing routine itself, though; it's the best one I've found so far for this picture format. I have used PrintWiz by Allen Macroware, which offers a variety of picture sizes, shapes, and graphics modes, but does not handle colors well. (Indeed, the documentation states that PrintWiz does not "stress color interpretation".) MAGIC LANTERN, on the other hand, not only has distinct shades from white to black for each of the four color registers available in the MicroIllustrator format, it also "sorts" the color registers according to luminance and color so that the brightest color in your picture will be white on the printout and the darkest will be black. The only other screen dump I have seen for MicroIllustrator pictures simply assigns white, black, light and dark grey to the color registers and leaves it to the user either to draw pictures that will print out acceptably or delve into the BASIC listing to change the color assignments. Clearly, MAGIC LANTERN's solution is the more "elegant".

The picture on the following page was printed with MAGIC LANTERN, which is available on several local BBSs and from the MACE disk library. Direct comments to the authors via CompuServe (accounts listed above).



PAGE ZERO

by Ann McBain Ezzell

This month I'm going to tell you about some memory locations that you might find useful when writing your own programs or trying to understand someone else's. I'll start, of course, on page zero and work my way up.

16 ($10) POKMSK
Handles POKEY chip interrupts. Bit 7 set indicates the BREAK key is enabled; you can use this location along with 53774 ($D20E) to disable the BREAK key. Set up a subroutine and do a GOSUB to it after every GRAPHICS command:

```
100 IF PEEK(16)<128 THEN RETURN
110 POKE 16, PEEK(16)-128: POKE 53774, PEEK(16): RETURN
```

77 ($4D) ATRACT
Attract mode timer and flag. You have probably noticed that most programs will start cycling through screen colors if no keys are pressed for a while. This is built-in protection for your monitor or TV screen. ATRACT is set to 0 each time a key is pressed, then incremented every four seconds during the vertical blank. When the value reaches 127, it is set to 254 until attract mode is terminated (usually by a keypress). You can force your Atari into attract mode with a POKE 77,128 command. Alternately, you can prevent attract mode from taking over during a joystick-driven program by periodically executing a POKE 77,0 command.

82, 83 ($52, $53) LMARGN, RMARGN
Left and right text margin columns. These are initially set to 2 and 39, and only affect GRAPHICS 0 and the text windows. You can squeeze more characters into a logical program line (three physical lines) by setting LMARGN to 0. RESET returns to the default values.

84 ($54) ROWCRS
Current graphics or text screen cursor row, from 0 at the top of the screen to a possible maximum of 191 at the bottom (GR. 8). You can POKE 84 to set the cursor where you want it on the screen, or PEEK(84) to find the cursor.

85,86 ($55,$56) COLCRS
Current graphics or text screen cursor column, from 0 at the left side to a possible maximum of 319 at the right (GR. 8). The value is stored in low byte, high byte order, so location 86 will always be 0 in modes other than 8 (because the other graphics modes have a maximum of 160 columns, 0-159).

88,89 ($58,$59) SAVMSC
Holds the address of the beginning of screen memory in low/hi byte order. You will often see SAVMSC used in programs which load and save screen displays. You can also calculate the address of screen memory (PEEK(88) + 256*PEEK(89)) and use it as a reference to PEEK or POKE directly into screen memory.

106 ($6A) RAMTOP
The size in pages (1 page = 256 bytes) of RAM. In a 48K machine, this location will hold a value of 160. You can POKE RAMTOP with a lower number to set up a (more or less) safe area for such things as redefined character sets, player/missile graphics, and so on.

186,187 ($BA,$BB) STOPLN
The line where the program was stopped because of an error, BREAK, STOP or TRAP. You can use this number to write your own error messages.

195 ($C3) ERRSAVE
The number of the error which stopped or TRAPped the program. (See May and June '85 Journals for explanations of error codes.)

559 ($22F) SDMCTL
Direct Memory Access (DMA) enable. This register is used when setting up player/missile graphics. You can also use it to speed up your program execution by about 30% by setting it to 0, which will blank the screen until you restore its value (PEEK it first to make sure you reset it properly). It's a good idea, though, not to put in the blanking command until your program is completely debugged; pressing RESET to restore the screen display will also erase any error messages.

That's all I have room for this month - I hope to be back next time with more information about useful locations.

M. A. C. E.

MICHIGAN ATARI COMPUTER ENTHUSIASTS

P.O. Box 2785, Southfield, MI 48037

BULLETIN BOARDS: MACE 978-1685/MACE WEST 582-0657

MACE HOTLINE (Information - Voice Line) 882-7104

PRESIDENT
Kirk Revitzer
Detroit
882-9109
Compuserve: 70346,1642
BBS: 882-5909

VICE-PRESIDENT
Alva Thomas
Taylor
287-3512
Compuserve: 74065,334
BBS: 287-4824

TREASURER
Burt Gregory
Southfield
552-0273

PROGRAM COORDINATOR
Scott Garland
West Bloomfield
851-9453

CORRESPONDING SECRETARY
Sharie Middlebrook
Dearborn
581-5560
BBS: 582-0657

RECORDING SECRETARY
Dino Roggero
Redford
BBS: 531-1701

DISK LIBRARIAN
Dave Zappa
St. Clair Shores
773-8551
BBS: 771-4126

CASSETTE LIBRARIAN
Mike Landis
Clawson
589-1789
Compuserve: 72675,1023

MEMBERSHIP CHAIRMAN
Paul Wheeler
Detroit
538-3649

M.A.C.E. JOURNAL EDITOR
Ann McBain Ezzell
Birmingham
646-4455

NEXT MEETING: 7/16/85 7:00 PM

Southfield Pavilion--Ten & a Half Mile Road and Evergreen

M.A.C.E.
P.O Box 2785
Southfield, MI 48037

$20.00* FOR A ONE YEAR MEMBERSHIP
(*Payable to M.A.C.E.)

M.A.C.E. MEMBERSHIP APPLICATION

Name _____ Phone _____
Street _____
City _____ State _____ Zip Code _____
Company (if applicable) _____
System Description _____ Disk/Tape _____
Suggestions _____
I can help with . . . _____

( )New ( )Renewal
If a Renewal:
MACE #_____

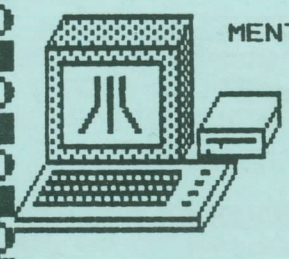
Date _____ Coupon _____
Amount _____ Membership Card Number _____
☐ Cash Expires _____
☐ Check Number

ACE 80 & ACE 80 XL
80 COLUMN CARTRIDGE

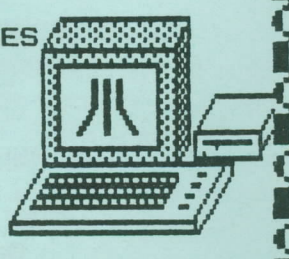
INTRODUCTORY OFFER - THROUGH JULY 31ST

ACE 80 (for Atari 800).....$49.95
ACE 80 XL (for Atari XL, 130 XE, 48K 400).....$49.95
Letter Perfect....(special).....$47.50
Data Perfect.....(special).....$47.50
Spell Perfect.....(special).....$47.50
Perfect Utiltity..(special).....$22.50

plus $2 shipping/handling **** Michigan add 4% sales tax



MENTION THIS AD FOR $5 OFF ACE CARTRIDGES



For more information contact
Michael Rudy
MAGIC SOFTWARE
3398 Seminole Trail
Monroe, MI 48161
(313) 289-1823

MICHIGAN ATARI COMPUTER ENTHUSIASTS
P.O. BOX 2785
SOUTHFIELD, MICHIGAN 48037

BULK RATE
U.S. POSTAGE
PAID
PERMIT #232
STER HTS, MI 48077

IMPORTANT DATED MATERIAL

PLEASE DO NOT DELAY